AMENDMENTS TO THE CLAIMS

Please amend the claims as follows.

1. (Currently Amended) A method for storing data blocks, comprising:

   storing a first data block and a second data block in a storage pool, wherein data stored in the storage pool is organized as a hierarchical tree, and wherein the first and second data block are stored on a first level of the hierarchical tree;

   obtaining a first data block location and a second data block location;

   calculating a first data block checksum for the first data block;

   calculating a second data block checksum for the second data block;

   storing a first indirect block at a first indirect block location on a second level of the hierarchical tree in the storage pool, wherein the first indirect block comprises a first block pointer comprising the first data block location and the first data block checksum and a second block pointer comprises the second data block location and the second data block checksum;

   calculating a first indirect block checksum for the first indirect block by applying a checksum function to the first indirect block; and

   storing a second indirect block at a second indirect block location on a third level of the hierarchical tree in the storage pool, wherein the second indirect block comprises the first indirect block location and the first indirect block checksum,

   wherein each of the first data block location, the second data block location, the first indirect block location, and the second indirect block location are separate physical locations in the storage pool.

2. – 4. (Cancelled)

5. (Previously Presented) The method of claim 1, further comprising:

   storing a birth value in a birth field in the first block pointer.

6. (Cancelled)

7. (Original) The method of claim 1, wherein the storage pool comprises at least one storage device.

8. (Original) The method of claim 1, wherein the storage pool is divided into a plurality of metaslabs.

9. (Original) The method of claim 8, wherein each of the plurality of metaslabs is associated with a metaslab ID.

10. (Previously Presented) The method of claim 9, wherein the first data block location comprises the metaslab ID and an offset.

11. (Previously Presented) The method of claim 1, wherein storing the first data block and the second data block comprises using a storage pool allocator.

12. – 17 (Cancelled)

18. (Currently Amended) A system for storing data blocks, comprising:

a storage pool configured to store data, wherein data stored in the storage pool is organized as a hierarchical tree, comprising:

a first data block, a second data block;

a first indirect block referencing the first data block and the second data block, wherein the first indirect block comprises a first data block checksum and a first data block location stored in a first block pointer, and a second data block checksum and a second data block location stored in a second block pointer;

a second indirect block, comprising a first indirect data block checksum and a first indirect block location; and

a storage pool allocator configured to store the first data block[[,]] and the second data block on a first level of the hierarchical tree, the first indirect block on a second level of the hierarchical tree, and the second indirect block on a third level of the hierarchical tree in [[a]] the storage pool,

wherein each of the first data block location, the second data block location, the first indirect block location, and the second indirect block location are separate physical locations in the storage pool.

19. (Cancelled)

20. (Previously Presented) The system of claim of claim 18, further comprising:

a data management unit configured to assemble the first indirect block and request the storage pool allocator to store the first indirect block.

21. (Original) The system of claim 20, wherein the storage pool comprises at least one storage device.

22. (Original) The system of claim 20, wherein the storage pool is divided into a plurality of metaslabs.

23. (Original) The system of claim 22, wherein each of the plurality of metaslabs is associated with a metaslab ID.

24. (Previously Presented) The system of claim 23, wherein the first data block location comprises the metaslab ID and an offset.

25. (Currently Amended) A computer system for storing data blocks, comprising:

a processor;

a memory;

a storage device; and

software instructions stored in the memory for enabling the computer system under control of the processor, to:

store a first data block and a second data block in a storage pool, wherein data stored in the storage pool is organized as a hierarchical tree, and wherein the first and second data block are stored on a first level of the hierarchical tree;

obtain a first data block location and a second data block location;

calculate a first data block checksum for the first data block;

calculate a second data block checksum for the second data block;

store a first indirect block at a first indirect block location on a second level of the hierarchical tree in the storage pool, wherein the first indirect block comprises a first block pointer comprising the first data block location and the first data block checksum and a second block pointer comprises the second data block location and the second data block checksum;

calculate a first indirect block checksum for the first indirect block by applying a checksum function to the first indirect block; and

store a second indirect block at a second indirect block location on a third level of the hierarchical tree in the storage pool, wherein the second indirect block comprises the first indirect block location and the first indirect block checksum,

wherein each of the first data block location, the second data block location, the first indirect block location, and the second indirect block location are separate physical locations in the storage pool.

26. (Currently Amended) A network system having a plurality of nodes, comprising:

a storage pool, configured to store data, wherein data stored in the storage pool is organized in a hierarchical tree, comprising:

a first data block, a second data block,

a first indirect block referencing the first data block and the second data block, wherein the first indirect block comprises a first data block checksum and a first data block location stored in a first block pointer, and a second data block checksum and a second data block location stored in a second block pointer;

a second indirect block, comprising a first indirect data block checksum and a first indirect block location; and

a storage pool allocator configured to store the first data block[[,]] and the second data block on a first level of the hierarchical tree, the first indirect block on a second level of the hierarchical tree, and the second indirect block on a third level of the hierarchical tree in [[a]] the storage pool,

wherein each of the first data block location, the second data block location, the first indirect block location, and the second indirect block location are separate physical locations in the storage pool,

wherein the storage pool is located on any one of the plurality of nodes, and

wherein the storage pool allocator is located on any one of the plurality of nodes.

27. (Currently Amended) A method for retrieving data in a first data block, comprising:

obtaining a first indirect block comprising a second indirect block location and a second indirect block checksum, wherein the first indirect block is stored at a first indirect block location on a third level of a storage pool, wherein data stored in the storage pool is organized as a hierarchical tree;

obtaining the second indirect block from a second level of the hierarchical tree using the second indirect block location, wherein the second indirect block comprises[[ing]] a first data block pointer and a second data block pointer, wherein the first data block pointer comprises a first data block location and a first data block checksum, and the second data block pointer comprises a second data block location and a second data block checksum, wherein the first and second data block locations are on a first level of the hierarchical tree;

applying a checksum function to the second indirect block to obtain a first calculated checksum,

comparing the first calculated checksum with the second indirect block checksum;

obtaining the first data block using the first data block location when the first calculated checksum equals the second indirect block checksum;

applying the checksum function to the first data block to obtain a second calculated checksum;

comparing the second calculated checksum with the first data block checksum; and

retrieving the data from the first data block when the second calculated checksum equals the first data block checksum,

wherein each of the first data block location, the second data block location, the first indirect block location, and the second indirect block location are separate physical locations in the storage pool.